Process for SharePoint Development

Information contained in this document is based on a SharePoint medium to large farm. It is recommended that a Solutions Deployment Path be put in place with a structured deployment path that includes the following teams.

- Developers
- Engineering
- Operations
- Help Desk
- Customer Outreach

The SharePoint Engineering Team should be kept informed throughout the development process including but not limited to code review(s).

Operations Team should be kept up to date on product lifecycle and impacts to system through the engineering team. In turn the operations team should keep the help desk informed on type of deployment (global vs. local), time lines and impact to system.

The above process diagram shows a high level view of the customizations analysis process

The steps listed above are the communications path between the development group and engineering.

Submitted for Review
After the Business Unit has gathered customer requirements the teams must meet to ensure awareness of customization

Review Customizations Design
Developer and Engineering teams meet to discuss but not limited to reusable code, Business Data Connector, security, search, site definitions, WCM scenarios, deployment using .WSP, possible upgrade issues.

Review Customization Code
Reviewing custom code allows for engineering and operations to understand performance and security requirements for IIS, .NET Framework and SharePoint Cache. This is a simple walk through of the code.

- Naming conventions
- Coding standards
- Optimizing performance
- Error handling

Coding Best Practice Checking
This is a Microsoft best practice check list to ensure that all team members agree and have performed the necessary steps to deploy custom code.

- Using SharePoint data and objects efficiently
- Performance concerns related to folders, lists, and SPQuery objects
    - Definition: In Microsoft Sharepoint, the SPQuery class is used to build query strings to filter / retrieve specific set of data programmatically from a SP List. The query strings created this way does not resemble a SQL syntax as much as it resembles a XSLT query object, where everything gets formatted as XML strings
- Writing applications that scan to large number of users
- Using Web controls and timer jobs
- Disposing of SharePoint objects
- Code cache the correct types of objects
- Thread synchronization, when necessary

Code Access Security (CAS) Verification
Default Security Permissions in Windows SharePoint Services
Windows SharePoint Services defines two security permissions by default as part of the **Microsoft.SharePoint.Security** namespace located in the

Microsoft.SharePoint.Security.dll. Each permission contains one or more attributes as follows:

SharePointPermission. Controls rights to access resources used by Windows SharePoint Services.

Attribute	Description
ObjectModel	Set to TRUE to use the Microsoft.SharePoint object model
UnsafeSaveOnGet	Set to TRUE to save data on HTTP-GET requests
Unrestricted	Set to TRUE to enable all rights associated with this permission.

WebPartPermission. Controls rights to access Web Part resources

Attribute	Description
Connections	Set to TRUE to participate in Web Part to Web Part communications
Unrestricted	Set to TRUE to enable all rights associated with this permission.

ASP.NET and SharePoint Security Policies
You can specify a level of trust that corresponds to a predefined set of permissions for ASP.NET applications. By default, ASP.NET defines the following trust levels:

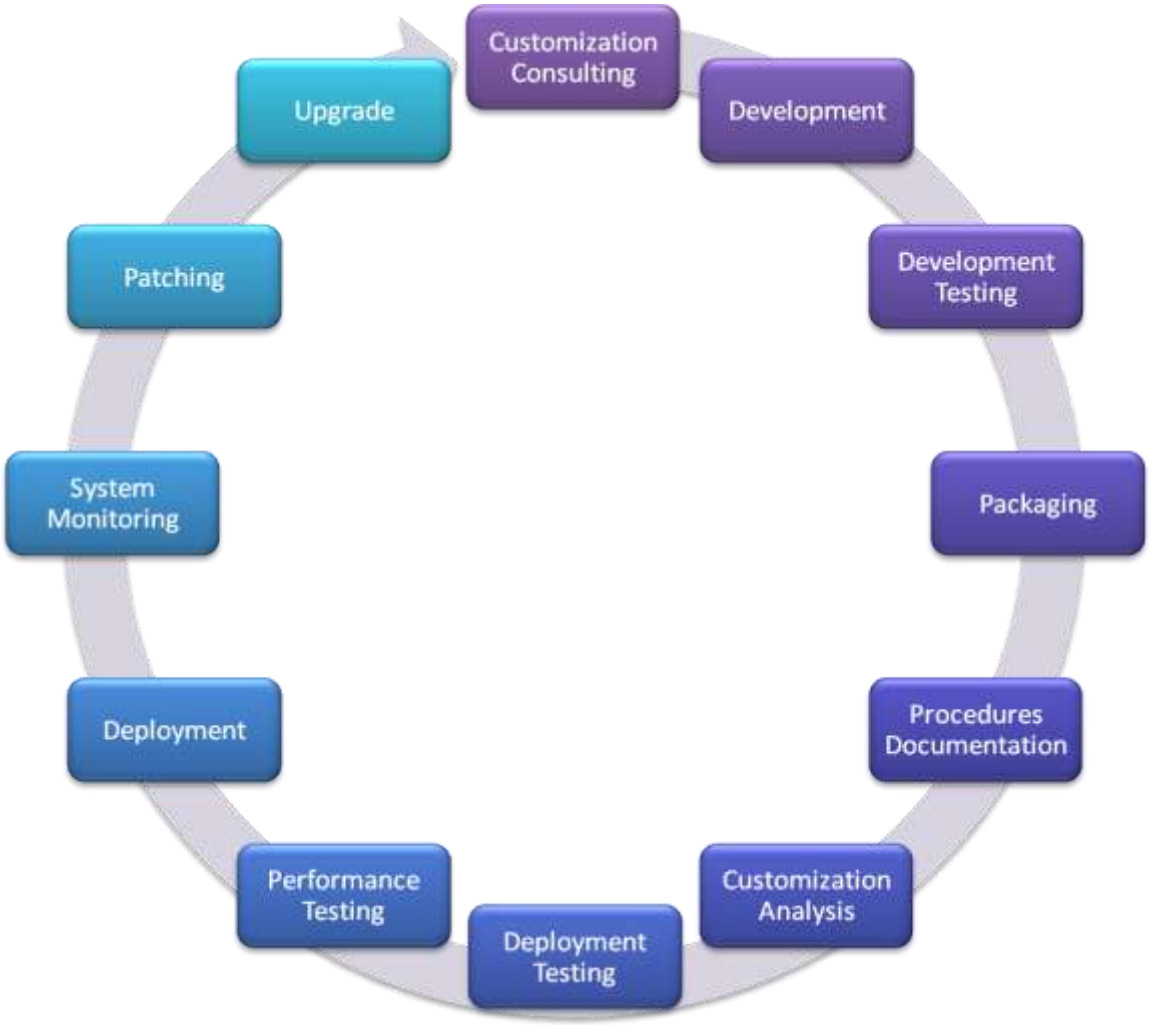
- Full
- High
- Medium
- Low
- Minimal

With the exception of the **Full** trust level, all trust levels grant only partial trust to the application folder of a virtual server instance. For more information on the ASP.NET trust levels, see Code Access Security for ASP.NET.

Additionally, Windows SharePoint Services defines two trust levels of its own:

- WSS_Minimal
- WSS_Medium

Review Analysis Results
Is the product ready for prime time?



The following process diagram is a high level overview of the customization development and deployment process

Each environment has an individual (Gate Keeper) that is the dedicated overall administrator for accountability and cleanliness, as well as knowledge of changes. Moving SharePoint Solutions from one environment to another mush follow a change approval process and board.

Key benefits of this model include but not limited to:

- Environment stability and dependability across all farms

- Quality assurance and testing will ensure stability and consistency in all environments

Service Strategies

Service strategies define the customization by which SharePoint can be modified. Supported customization type is shown in Table 1. If a customization type is not listed, it might still be supported.

Table 1. Supported Customization Types

Customization Type	Supported	Comment
Custom-managed path	Yes/No	Current managed path have been set for each NASA Center. Any additional requirement for a managed path must be approved by the system owner or higher.
SharePoint database schema change	No	Changes directly to the database schema are not supportable.
SharePoint database data access	No	No access will be provisioned directly to the SharePoint databases. All access to the databases will be via the object model (OM) or Web services.
Modify built-it SharePoint files	No	No support any changes to the files on the file system. This includes shipped site definitions.
Web services access	Yes	All Web services that ship with SharePoint are exposed.
SharePoint designer editing	Yes	Any change that can be made with SharePoint Designer will be supported.
Change mapped host name after deployment	No	No support the changing of the hosted domain name, i.e., sharepoint.nasa.gov after the environment has been deployed
Visual Studio installed on deployment	No	Visual Studio to be installed on a WFE server for troubleshooting.
Install Office client on server	No	Office clients to be installed on a WFE server.

Customization Type	Supported	Comment
SharePoint solution	Yes	Solution packages are a deployment and packaging mechanism for SharePoint customizations.
SharePoint feature	Yes	SharePoint features are server-side file system level customizations that can contain modular items that can be installed and activated in a SharePoint environment.

Customization Type	Supported	Comment
Feature stapling	Yes	Feature stapling, also known as feature site template associations, is a capability for a feature to be attached to all new instances of sites that use a given site definition without modifying the site definition or creating code routines to activate the feature on each site.
Feature event receiver	Yes	A feature event receiver is a server-side code routine that is called as part of certain key events in the lifetime of a feature, such as installation, activation, deactivation, and removal.
Window server service	No	A service installed on the server to perform major actions. It runs in a separate process space from the SharePoint instance. A service has a relatively high cost for deployment and manageability.
Timer job	Yes	A timer job is a built-in SharePoint object structure that can perform various tasks within the SharePoint environment on a scheduled or one-time event basis. Timer jobs pose a risk for platform performance and stability.
Scheduled Task	Yes	A scheduled task permits the execution of code on the Windows server on a defined schedule. On event occurrence, the code is executed. Scheduled tasks run outside of SharePoint and are independent of the application. There is no trigger from within SharePoint.
Web application	No	Application code or content installed in a new Web application directory or as a sub-directory under an existing Web application, which provides non-SharePoint functionality. Additional Web applications pose an additional burden on operations to support.
Web service	Yes	A Web service is server-side code that exposes code routines to be called from remote systems.
Site definition	No	A site definition is a server-side collection of files that defines the structure of one or more site templates. Once a site is created with a custom site definition, it is dependent on that site definition going forward, including when upgrading between versions of SharePoint.
List definition	Yes	A list definition provides the schema for a SharePoint list.
Site template	Yes	A site template is a package that contains a customized site design based on an existing site definition. A site template should not be confused with a site template that is part of a site definition.
List template	Yes	A list template is a package that contains a set of customizations from a base list structure. A list template should not be confused with a list definition.
Field types	Yes	Field types are an extensible method for allowing new common data structures within SharePoint for use in lists or other locations that can use SPField objects.
Content type	Yes	A content type is a reusable collection of settings that can be applied to a certain category of content. Content types enable the management of metadata and behaviors of a document, item, or

Customization Type	Supported	Comment
		folder type in a centralized, reusable way.
Column template	Yes	A site column, also known as a field, is a reusable column definition, or template, that can be assigned to multiple lists across multiple SharePoint sites. Site columns decrease duplication of work and help ensure consistency of metadata across sites and lists.
Delegate control	Yes	Delegate controls allow certain predefined pieces of the SharePoint environment to be replaced by other controls through SharePoint features.
Form template	Yes	Form templates are ASCX-based control templates that bring together multiple controls into a nested structure, where each control consists of templates and can be used to richly extend list item forms.
Custom action	Yes	Custom actions are Items added to menus or toolbars through the use of features.
_Layouts page	Yes	_Layouts pages are any page stored in the _layouts virtual directory within SharePoint.
Event handler	Yes	An event handler is a server-side code routine that is called when registered events occur within the SharePoint environment. Custom event handlers can be attached to and have a scope of Web application, site collection, Web, list, or document library.
Backward-compatible event handler	Yes	A backward-compatible event handler is a server-side code routine that is called when registered events occur within a SharePoint document library. Backward-compatible event handlers can be attached to and have a scope of a document library only.
Coded workflow	Yes	Coded workflows are Windows Foundation workflows that reference an assembly for the workflow pipeline.
No-code workflow	Yes	No-code workflows, also known as declarative workflows, are sequential workflow pipelines, configured on a list or document library. They require no installation of server-side code.
Workflow activity	Yes	Workflow activities are compiled classes that are used as a part of the modular steps in a workflow. An activity is used as part of a workflow action that will be executed. Workflow activities run as server-side code when used as activities within SharePoint.
Workflow condition	Yes	Workflow conditions are compiled classes that are used as a part of the modular steps in a workflow. A condition is used to determine when a workflow action will be executed. Workflow conditions run as server-side code when used as conditions within SharePoint.
Web Part	Yes	A Web Part is a modular component that is used within SharePoint to perform activities or display information.
SharePoint Theme	Yes	SharePoint Themes are collections of graphics and cascading style sheets that can modify how a Web site looks.

Customization Type	Supported	Comment
Document icons	Yes	Document icons are the graphical items used to represent documents exposed within document libraries. Each preconfigured document type in SharePoint has a corresponding document icon entry in a server configuration file, but other file formats may lack the appropriate icon.
iFilter and protocol handlers	Yes	An iFilter is a server-side component that is used by the indexing system to index documents that are identified as a file format that is associated with the iFilter. Custom iFilters require deployment to all servers in the environment that do indexing and require changes to mapped properties for search.
Document converter	Yes	A document converter is a custom executable file that takes a document of one file type, and generates a copy of that file in another file type.
Information management policy	Yes	An information management policy is a set of rules for a certain type of important content. Policy enables administrators to control and evaluate who can access the information, how long to retain information, and how effectively people are complying with the policy itself.
Business Data Catalog (BDC) definition	Yes	A BDC Definition is an XML file that defines the structures used by the BDC.
BDC-based search crawl	No	Search crawls based on a BDC definition will not be supported. Properties associated with a user profile will be indexable and displayed within People Search.
Excel User-Defined Function (UDF)	Yes	User-Defined Functions (UDFs) are server-side managed code, executed by Excel Calculation Server to provide capabilities not included in the base product. These include: • Functions that are not built into Excel. • Custom implementations to built-in functions. • Custom data feeds for legacy or unsupported data sources, and application-specific data flows.
InfoPath Form custom code	Yes	Custom code included within an InfoPath Form that is uploaded and hosted by the server-side InfoPath Forms Server. Subject to custom code review process.
InfoPath Form view control	Yes	Server-side InfoPath Form viewer control that can display a Web browser view of an InfoPath Form.
HTTP handler	No	HTTP handlers are used to handle file requests within the ASP.NET 2.0 framework. For example, within ASP. NET 2.0 and SharePoint 2007, ASPX page and ASMX Web service requests are served by HTTP handlers.
HTTP module	Yes	HTTP modules are classes that handle runtime events in ASP.NET 2.0 and SharePoint 2007.

Customization Type	Supported	Comment
Pluggable authentication provider	No	A pluggable authentication module used by ASP.NET 2.0 and SharePoint to replace the built-in authentication providers. Custom modules require a high level of testing to ensure security.
Pluggable Single Sign-On Provider	No	A Single Sign-On Provider is a modular component that can be used to handle the storage and mapping of credentials for use in connecting with third-party or backend systems. Requires high level of security review and scrutiny.
STSADM command extension	No	A custom STSADM command extension is server-side code that allows STSADM to have new commands available to execute from the command line.
Inline code	No	Inline code is server-side code that is directly embedded into a Web page. Inline code has a high level of risk compared to compiled assemblies.
Web.config settings changes	Yes	The Web.config file controls most settings for the ASP.NET environment that SharePoint 2007 is built on. Typically this is modified as a part of a solution. Changes should be scripted as a part of the custom solution installation and configuration instructions. Manual modification of the web.config settings is not supported.
Security policy	Yes	A custom security policy grants Code Access Security (CAS) rights to certain server-side code components running under a specific Web application context.
COM server	No	A COM server is an unmanaged library that typically executes on a server with full permissions. COM objects typically perform poorly when interops are set up between the COM object and .NET code.

Source Control and Build Automation

Whether a project includes one or many developers, source control is very important for SharePoint. SharePoint doesn't support the nature of versioning (see following sections) so management of the source and build process is vital for a successful project. Microsoft recommends Team Foundation Server for source control and build automation. Tools like WSPBuilder can greatly assist in the build process for your solutions.

Customers can use this checklist to make sure they have proper documentation ready when proposing custom SharePoint Online Dedicated solutions.

The following documentation is required for High Level Design (HLD) review.

High Level Design Document (In House Customization)

- ☐ Document: Use the provided template or another document format to indicate the requirements and planned design and implementation details for your customization.

High Level Design Document (Third-Party Product)

- ☐ Document: Use the provided template or another document format to indicate the requirements and planned design and implementation details for the third party customization. Outline what you plan to test (memory usage, scale, and functionality). It is important that you have a validated the third-party product prior to providing the HLD to Microsoft.

Customization Documentation Checklist

The following documentation is required for customization analysis and testing and should be included with the customization package.

Installation Instructions

- ☐ Document: How to install solution package or customization (not third-party documentation).
- ☐ Document: How to smoke test that the package was properly installed and the custom solution works correctly.
- ☐ Document: Rollback plans (uninstall, clean up, and fix anything) when the customization is removed.
- ☐ Reference: Any architecture diagrams that detail dependencies, data flow, etc.
- ☐ Document: Any third-party licensing keys or certificates for deployment to the environment.
- ☐ Document: Customer Security and Compliance Review Approval of the design.
- ☐ Document: Monitoring guidance detailing logging details and instrumentation for the custom solution.

Test Documents and Results

- ☐ Document: Test Plan – What was tested, why it was tested, and how it was tested. Detail what assumptions were made. A usage profile or any data points

that describe the test environment should be detailed here. Were multiple WFEs tested?

- ☐ Document: Unit Test Results - What unit tests were run for feature or functionality testing? What clients were used? What were the results? What dependencies were tested? What failure situations were tested?

- ☐ Document: Performance/Scale Test Results - What performance or scale tests were run against the customization? What was the usage profile used? What assumptions were made?

Dependencies

- ☐ Document: List of all dependencies for the code (Web service, account, database, solution/feature, patch, tool set, library).

Event Error Codes

- ☐ Document: List of all Event entries generated by the customization with Event IDs.

- ☐ Document: Create troubleshooting instructions for all Event entries (error code, severity, root cause).

Client Troubleshooting Documentation

- ☐ Document: Troubleshooting guidance for client-facing problems.

- ☐ Account: One or two test accounts if appropriate for diagnosing problems with related dependencies. The account should be a low-privileged account used for no other purpose.

Source Code

- ☐ Project Files – Required where available.

Solution Package

- ☐ Binary: Final tested code in a customization package for deployment.

- ☐ Scripts: Install and uninstall scripts for the customization package.

- ☐ Scripts: Install and Uninstall scripts for the customization package in the Secondary Data Center (If warranted)

Update for existing customizations

- ☐ Document: Summary of changes between current version and previous version. Provide adequate detail to tell what has changed.
- ☐ Document: Rollback plan to install previous version of the customization.
- ☐ Document: If appropriate, uninstall information for previous customization.
- ☐ Document: Detail any expected problems that will be encountered during upgrade.

Development Criteria

With the use of MS Team Foundation Server and MS Visual Studio the development team has the ability to meet all criteria for moving solutions through the various gates in a timely matter. As part of the installation package the following will be sumitted to the engineering staff.

Results from:

- IntelliTrace
- Code Analysis
- Test Impact Analysis, Coded UI Test

By providing this information both the engineering and development staff will be able to identify test impact from code changes, profile application performance and generate realistic test data.

Operations Criteria

Supported by the engineering staff, operations will be responsible for day-to-day systems/services monitoring. Using DocAve and the operating system performace tools, additional tools maybe required as the farm grows. It is also important that operations monitors Helpdesk ticket trends for site collection performance concerns.

DocAve provides real time monitoring for Farm Topology, Services, CPU/Memory, Networking, Index, and Environment Search. Real time reports can be exported ti datasheet and/or screen shots. Reports should be generated at end of month and reported to the engineering staff.

Infrastructure reports include Differnece, Storage trends, SharePoint alerts and load times for site collection. Load times for site collections should be utilized in conjuction with Helpdesk ticket trends.

Systems Monitoring
The engineering team will provide the operations team with monitoring criteria. Monitoring of the serverices should be conducted on a ramdom schedule every other month as to no impead performance. Based on data gathering the most utilized times are Monday's and Tuesday's with the montly activity heavy the first two weeks of the month and tailing off the last week.

In addition to DocAve reporting baseline OS monitoring should be conducted on a monthly bases. OS monitoring will not only assist in scaling but determining the root cause of performance issues of the investigation phase.

Basic OS performance short list:

Front-end subsystem

Category	Details
Memory	Baseline for memory usage is 50-60 percent of physical memory, as measured by the Memory/Abailable Bytes counter.
CPU	Utilization and spikes in CPU usage providey early indication of performance issues. Baseline for this category is 30-50 percent average usage.
Concurrnet Connections	Target 150-250 average concurrent connections per server.

Back-end subsystem

Category	Details
Memory	A healthy CPU usage for back-end servers at 30-50 percent average utilization.
CPU	CPU utilization is not as important for back-end servers.

	Baseline for this category is 30-50 percent average usage.
Disk I/O	Disk queue length, as measured by the PhysicalDisk(_Total)/Current Disk Queue Length counter, in addition to disk I/O, is extremely important for back-end servers.
Concurrent connections/SQL Server blocking	For back-end servers, the number of concurrent connections is relevant, especially when they correspond to instances of SQL Server blocking. Correspondingly, Microsoft IT tracked the baseline of SQL Server blocking by using the SQLServer:Locks(_Total)\Number of Deadlocks/sec counter. An acceptable number of deadlocks is below one per second.
Percentage of database fragmentation	The baseline fragmentation is below 8 percent on back-end servers.

Basic OS performance long list:

Front End Perf Counters

\\host name\Processor(_Total)\% Processor Time

\\host name\Process(LSASS)\% Processor Time

\\host name\Process(w3wp)\% Processor Time

\\host name\Process(OWSTIMER)\% Processor Time

\\host name\Memory\Pages/sec

\\host name\Memory\Pages/sec

\\host name\Memory\Available Bytes

\\host name\System\Context Switches/sec

\\host name\Process(w3wp)\Working Set

\\host name\Process(w3wp)\Private Bytes

\\host name\Process(w3wp)\Page Faults/sec

\\host name\Process(w3wp)\Working Set

\\host name\Process(w3wp)\Working Set Peak

\\host name\Process(w3wp)\Virtual Bytes

\\host name\Process(w3wp)\Virtual Bytes Peak

\\host name\Process(w3wp)\Private Bytes

\\host name\Process(w3wp)\Page File Bytes

\\host name\Process(w3wp)\Page File Bytes Peak

\\host name\Process(OWSTIMER)\% Processor Time

\\host name\ASP.NET\Request Execution Time

\\host name\ASP.NET\Request Wait Time

\\host name\ASP.NET\Requests Queued

\\host name\ASP.NET\Requests Rejected

\\host name\ASP.NET\Worker Process Restarts

\\host name\ASP.NET\Application Restarts

Back End (SQL Server) Performance Counters

\\host name\\Processor(_Total)\% Processor Time

\\host name\Memory\Pages/sec

\\host name\System\Context Switches/sec

\\host name\Process(sqlservr)\% Processor Time

\\host name\Process(sqlservr)\Working Set

\\host name\Process(sqlservr)\Private Bytes

\\host name\SQLServer:General Statistics\User Connections

\\host name\SQLServer:Databases\Transactions/sec

\\host name\SQLServer:Locks(_Total)\Number of Deadlocks/sec

\\host name\SQLServer:Locks(_Total)\Lock Waits/sec

\\host name\SQLServer:Locks(_Total)\Lock Wait Time (ms)

\\host name\SQLServer:SQL Statistics\Batch Requests/sec

\\host name\PhysicalDisk(_Total)\Current Disk Queue Length

\\host name\PhysicalDisk(_Total)\Disk Read Bytes/sec

\\host name\PhysicalDisk(_Total)\Disk Write Bytes/sec

Environments:

Developer

Developer environments will reside on the developer's workstations, this maybe individual or shared team environment, with both MS SQL and SharePoint residing in the same VM.

For developers, Windows SharePoint Services Dev Central should promote code and knowledge sharing and consequently maximize development efficiencies and lower maintenance costs. Windows SharePoint Services is unlikely to be the best solution to maintain code—developers should use an application that specializes in code versioning and sharing support, such as Team Foundation Server. But Windows SharePoint Services can support all of the development activity regarding the code. The Windows SharePoint Services Dev Central site should also clarify requirements for development on Windows SharePoint Services infrastructure.

- What are the requirements for development of Windows SharePoint Services components in your organization? What standards should be followed?
- What are the requirements for the deployment of Windows SharePoint Services components? How do we consistently deploy and update Web parts, site definitions, features, and templates?
- What are the standards for security? For example, what are the standards for what can be added to the Microsoft .NET Framework Global Accessory Cache (GAC) or Trust Levels?
- How are features and solution deployments tested? What is the test environment?
- What documentation is required and where should the documentation be stored?
- How are warnings and errors logged and tracked?
- What types of logging are performed by custom components?

- How are issues raised and tracked?
- What are the responsibilities for ongoing code support?

As with applications, sites, templates, and other components, Windows SharePoint Services Dev Central should maintain a catalog of custom components. The catalog should explain installation and support issues that are related to each component, including code dependencies.

Additionally, the Windows SharePoint Services and Community Site should maintain a single issue-tracking application that you can use to highlight problems, communicate about those problems, and announce resolutions. By locating this application in the same site collection as the catalogs, you can easily tie issues to specific components of your Windows SharePoint Services implementation.

Integration

All developer work comes together before going to the test environment.

This should be a clean, stable and dependable environment. This does not need to be a complete farm and may contain both MS SQL and SharePoint on the same system and may reside in the Operations ESX farm. Developers have no administrative permissions to operating system, and only Visual Studio remote debugger installed. The SharePoint technical lead or lead architect will control environment and facilitate the integration of the various developer efforts.

Gate Keeper: Technical Lead and/or Lead SharePoint Architect

Key Point of integration of updates coming from various developers on the team

- Gatekeeper can better hold developers accountable to well unit tested code because of tighter control and closer eye on changes and integration points.
- The gatekeeper is responsible to develop the implementation of steps for production launches, updates and the beginnings of any documentation for updates starts at Gate 1.

Quality Assurance (Staging)

No developer access to ensure quality and reliable tests

A good production roll-out should have properly documented implementation steps, and SharePoint solution packages must be used. This is where the Operations staff should be performing a “dry-run” of the production roll-out steps.

Gate Keeper: Operations Lead

- First line of defense production roll-outs, and the first place where CAB is necessary. The technical lead, lead architect, and or project manager should mutually sing-off on any updates to the QA environment.
- Production lead will perform all updates that get approved
- Rough draft of implementation steps will be provided as a deliverable to the project manager during the CAB approval request
- Properly documented implementation steps and back out process
- Deploy in SharePoint Solution Packages

Production

Prior to going in to production a solution should pass through two change control processes. First being the normal change control process as outlined in the SharePoint Change Control Board process. The second should be a department wide change control to ensure support staff is prepared, all notifications have been sent out and proper processes are in place.

- Gate Keeper: Operations Lead
- Department wide CCB approval
- Final draft of implementation steps required
- Steps implemented by Operations

SQL

No changes should be made to the SharePoint SQL Schema outside of SharePoint Service Pack and Infrastructure updates and/or upgrades. Changes outside of Service Pack and Infrastructure upgrades will void any Microsoft Support.